VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Visualizace Dat ve Formátu VCF

Visualisation of Data in VCF Format

2020 Martin Kurfürst

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **Martin Kurfürst**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: Visualizace Dat ve Formátu VCF
Visualisation of Data in VCF Format

Jazyk vypracování: čeština

Zásady pro vypracování:

Cílem práce je implementace programu s grafickým uživatelským rozhraním umožňujícím zobrazit vybrané sekvence lidského genomu. V rozhraní budou dále zobrazeny genetické mutace dodané ve formě souborů VCF. Program bude umět filtrovat a porovnávat specifické VCF soubory a vizualizovat časové řady vývoje genetických mutací.

1. Popis formátu VCF a formátů pro práci s DNA.
2. Základní popis nástrojů použitých při generování VCF.
3. Popis implementace a použitých nástrojů.
4. Experimentální ověření programu na vybraných VCF souborech.

Seznam doporučené odborné literatury:

[1] VCF Specification, online: http://samtools.github.io/hts-specs/VCFv4.3.pdf, 2016.
[2] Nielsen, Rasmus and Paul, Joshua S and Albrechtsen, Anders and Song, Yun S, Genotype and SNP calling from next-generation sequencing data". Nature Reviews Genetics, 2011.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Michal Vašinek, Ph.D.**

Datum zadání: 01.09.2019
Datum odevzdání: 30.04.2020

doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry

prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 15. května 2020 .....

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.

V Ostravě 15. května 2020 .....

Rád bych poděkoval Ing. Michalu Vašinkovi, Ph.D za vstřícnost a cenné rady při vytváření této bakalářské práce.

Abstrakt

Hlavním cílem bakalářské práce je vytvoření aplikace, která bude schopna zobrazit vybrané sekvence genomu, varianty genetických mutací a jejich časové řady vývoje. První část práce obsahuje popis základních pojmů genetiky, sekvenování genomu, samotnou specifikaci VCF formátu, který se používá pro uložení genetických mutací, a přehled ostatních formátů pro práci s DNA. Ve druhé části je popsána samotná implementace programu pro zobrazení. Práce je zakončena testováním aplikace.

Klíčová slova: VCF, DNA, sekvenování

Abstract

The main goal of the bachelor's thesis is to create an application that will be able to display selected genome sequences, variants of genetic mutations and their time series of evolution. The first part of the thesis contains a description of the basic concepts of genetics, genome sequencing, the specification of the VCF format, which is used to store genetic mutations, and an overview of other formats for working with DNA. The second part describes the implementation of the display program itself. The work ends with testing the application.

Keywords: VCF, DNA, sequencing

Obsah

Seznam použitých zkratek a symbolů

ASN.1	–	Abstract Syntax Notation One
BAM	–	Binary Alignment Map
BCF	–	Binary Call Format
BED	–	Browser Extensible Data
DNA	–	Deoxyribonucleic Acid
FASTA	–	Fast Alignment
GFF	–	Gene-Finding Format
GTF	–	Gene Transfer Format
NGS	–	Next-Generation Sequencing
RNA	–	Ribonucleic Acid
SAM	–	Sequence Alignment Map
UTF-8	–	Unicode Transformation Format - 8
uBAM	–	unmapped Binary Alignment Map
VCF	–	Variant Call Format

Seznam obrázků

Seznam výpisů zdrojového kódu

Úvod

V dnešní době, kdy je technologie na vzestupu, nezůstává pozadu ani bioinformatika. Od přelomu tisíciletí, kdy byl kompletně osekvenován lidský genom, se tato oblast rozrůstá, čím dál rychleji. Na tyto získaná data existují nástroje, pomocí nichž se dají zjistit nemoci zakódované v samotné DNA. Abychom byli tyto nemoci schopni identifikovat je potřeba analyzovat osekvenovaný genom. Při analýze probíhá mnoho úkonů, jedním z nich je detekce variant. Osekvenovaný genom se zarovná s referenčním genomem a poté se vyhledávají sekvence, které se liší a jsou zapsány do VCF souboru. Tyto odlišnosti nazýváme varianty. A právě na VCF soubory se práce zaměřuje.

Nejprve je nutno popsat základní pojmy bioinformatiky, se kterými se lze setkat v tomto odvětví. Každý formát má určitou specifikaci a ani VCF formát není výjimka. V další kapitole je tudíž vylíčena základní specifikace VCF formátu. Kapitola taktéž obsahuje nástroje pro generování VCF souboru. Pro práci s DNA se nepoužívá pouze VCF formát, ale existuje celá řada formátů, které se této problematice věnují. Přehled formátů z odvětví DNA popisuji v další kapitole.

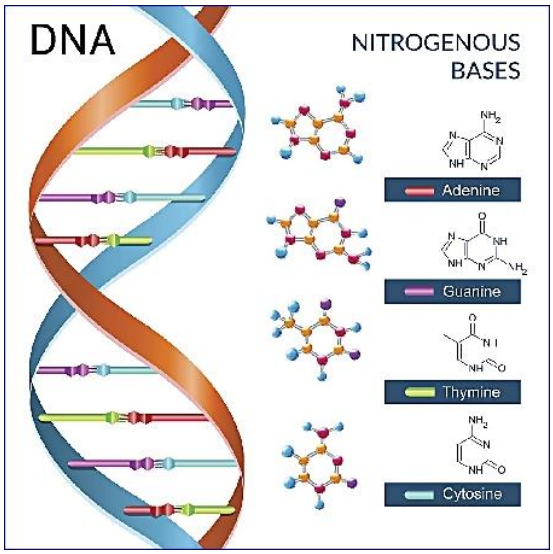
Hlavním cílem práce je vytvořit aplikaci, která bude schopna zobrazit konkrétní sekvenci bází referenčního genomu, detailní přehled variant z VCF souborů a a porovnaní genových mutací na časové řadě vývoje. Touto částí se zaobírám v druhé polovině práce, kde jsou popsány nejdůležitější články implementace. V poslední kapitole najdeme samotné ověření programu na testovacích VCF souborech.

1 Genom

Každý buněčný organismus v sobě nese genetické informace. Potomci jsou většinou podobni svým rodičům, což vyvozuje fakt, že po svých rodičích dědíme specifické vlastnosti. Každá buňka obsahuje soubor všech informací organismu, které nazýváme genom.[1][2][3][4] Tyto informace najdeme uložené uvnitř buněk, kde jsou zapsány v DNA. Pokud se budeme bavit o lidském genomu, tak musíme uvést, že většina informací se nachází v jádru buňky, ovšem menší část najdeme také v mitochondriích.

1.1 DNA

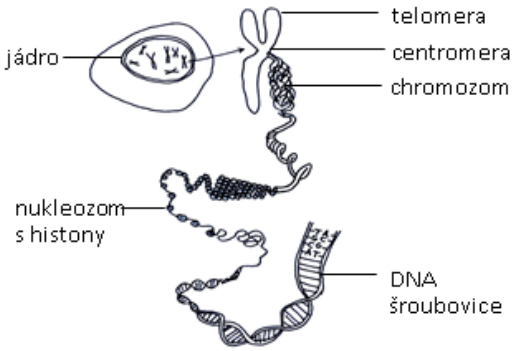
DNA je deoxyribonukleová kyselina, která slouží k uchování genetické informace.[5][6] Nukleovou kyselinu tvoří nukleotidy, které vznikají spojením cukerné složky, zbytku kyseliny trihydrogenfosforečné a dusíkaté báze (adeninu - A, thyminu - T, cytosinu - C a guaninu - G). Dané báze plní informační funkci. Stavební prvky, z nichž je DNA tvořeno, se v prostorovém uspořádání formují do útvaru nazývaného dvojitá šroubovice.



Obrázek 1: DNA [7]

1.2 Chromozom

Soubor všech informací neboli genom není uložen v jedné molekule DNA, načež je rozdělen do více úseků, které tvoří lineární útvary a nazýváme je chromozomy. Tyto úseky jsou složené z DNA a proteinů, které jsou důležitou součásti při spiralizaci a fungování DNA. Název pochází z řečtiny, ze slov *chroma* a *soma*, což v překladu znamená barevné tělísko.[8][9][10]



Obrázek 2: Chromozóm [11]

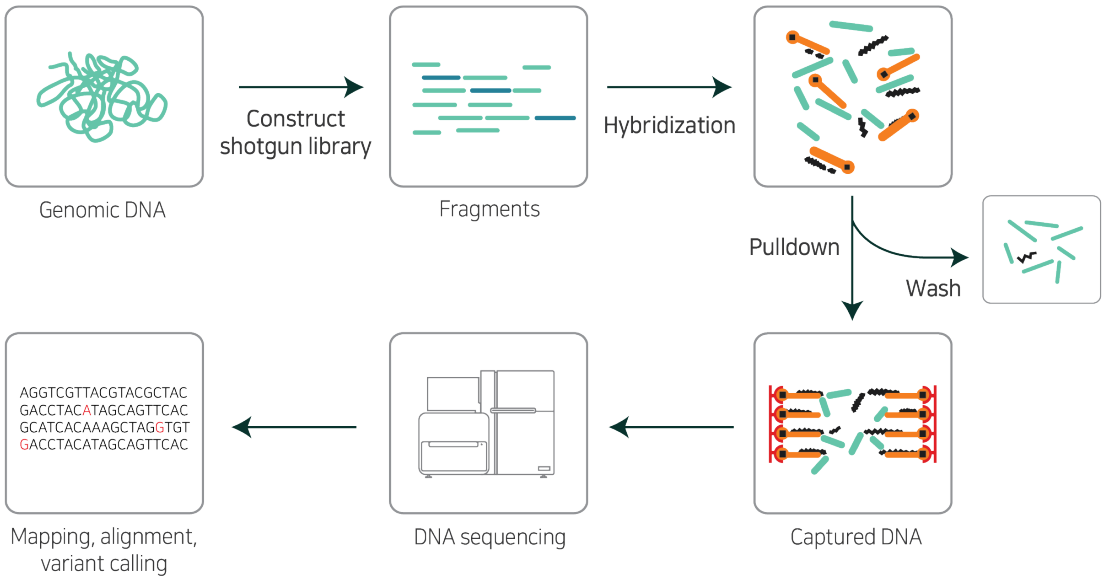
Chromozomy dále dělíme na dvě podskupiny - autozomy a gonozomy. Autozomy jsou takové chromozomy, které tvoří stejné páry a jejich přítomnost není specifická pro určité pohlaví. Oproti tomu gonomozy jsou pohlavní chromozomy a jejich stavba je odlišná. Lidský genom je rozdělen na 46 chromozomů, které se dělí na 23 párů.[9] Z těchto párů chromozomů jich je 22 autozomních a pouze 1 pár gonomozů, které označujeme X a Y.

1.3 Genotyp a fenotyp

Genotyp označuje veškerou genetickou informaci organismu a tyto informace určují o jaký druh organismu se jedná. Soubor vlastností a znaků organismu, které můžeme pozorovat se nazývá fenotyp. Zjednodušeně řečeno fenotyp je spojení genotypu a prostředí, ve kterém organismus žije.[12][13]

2 Sekvenování genomu

Termínem sekvenování genomu rozumíme zjišťování pořadí nukleových bází (A, C, G, T) v sekvencích DNA.[14] V těchto datech nazývané genetika je spousta informací pro lékařskou diagnózu, které odhalují spoustu heterogenních chorob. Na počátku sekvenování lidského genomu byli Paul Berg, Frederick Sanger a Walter Gilbert. Tito pánové umožnili pokrok v oblasti analýzy DNA. Technologie zvaná „Sangerovo sekvenování“, jež byla po jednom z nich pojmenována, znamenala vzestup sekvenování DNA, což umožnilo uvedení prvního automatizovaného sekvenceru DNA.[15][16] Vývoj postupoval díky rozhodujícím krokům v oblastech nanotechnologií a informatiky a proto došlo k nahrazení „Sangerova sekvenování“ nově vznikajícími technologiemi, jež jsou označovány jako sekvenování nové generace NGS.



Obrázek 3: Zobrazení průběhu sekvenování genomu [17]

2.1 Vytváření knihovny

Při práci s NGS se dodržuje postup, který můžeme vidět na obrázku 3. Nejprve se pro daný vzorek DNA připraví knihovna. Pojem knihovna označuje soubor fragmentů DNA /RNA, což představuje genom neboli cílovou oblast.[15] Ve většině případech se knihovna začíná vytvářet štěpením DNA/RNA na krátké úseky. V moment, kdy jsou fragmenty připraveny, je každý fragment označen adaptérovou sekvencí, který umožňuje dané vzorky navzájem odlišit.[18]

V dalším kroku je nutné obohacení, to můžeme chápat jako naklonování.[15][18][19] Tímto krokem zvýšíme množství cílového materiálu v knihovně, který má být sekvenován. Velké množství klonů, které bylo vytvořeno, je posláno do sekvenceru, kde se zjistí pozice bází.

2.2 Analýza

Nyní obrátíme pozornost na analýzu dat získaných během sekvenování. Analýzu rozdělujeme na tří části: primární analýza, sekundární analýza a terciární analýza.[15][19]

2.2.1 Primární analýza

Hlavní analýza se skládá z detekce a analýzy prvotních dat, cílení na generování čitelných sekvenčních čtení a vyhodnocování základní kvality. Mezi typické výstupní formáty patří FASTQ nebo nemapovaný soubor mapy binárních zarovnání uBAM. Při sekvenování dochází k nepřesnostem a tyto chyby jsou vyjádřeny ve skóre kvality [1]. Při kontrole kvality je nedílnou součástí formát FASTQ, ve kterém jsou obsaženy prvotní čtení sekvence, názvy souborů a hodnoty kvality. Jelikož je kvalita dosud nezpracovaných sekvencí velice důležitá pro úspěch analýzy, tak byly vytvořeny nástroje (QC-Chain, FastQC) pro hodnocení kvality nezpracovaných údajů. Jestliže sekvenční čtení má dostatečnou kvalitu, je možné srovnání s referenčním genomem. Skóre kvality je taktéž využíváno pro filtrování a ořezávání sekvencí. Na konci každého čtení se provede oříznutí, což odstraní adaptérovou sekvenci, která může narušit mapování a sestavování.[15][19]

2.2.2 Sekundární analýza

Dalším krokem je sekundární analýza, ve které se setkáme s mapování sekvencí, které má dvě možnosti, buď to jsou sekvence zarovnány s referenčním genomem nebo sestavení de novo, které vytváří sekvenci od úplného začátku. Sluší se dodat, že v klinické genetice se využívá mapovaní proti referenčnímu genomu.[15][19]

Seřazení sekvencí je v bioinformatice základní problém.[15][19] Při sekvenování genomu jsou generovány miliardy fragmentů DNA/RNA, které musejí být správně poskládány. Tento proces představuje obrovskou výpočetní výzvu, jelikož může nastat velké množství problémů. Nejčastějším vstupním souborem je FASTQ, pro výstup jsou netypičtější formáty SAM nebo BAM. Pro určení změn na genomu se používá detekce variant[2]. Zjednodušeně řečeno detekce variant je proces, kde se identifikují varianty ze sekvenčních dat. K tomuto účelu se používá mnoho nástrojů, vybral jsem pár z nich:

SAMtools, BCFtools

Jedná se o sadu nástrojů, které jsou určeny pro interakci a následné zpracování krátkých sekvencí DNA ve formátu SAM, BAM a CRAM. Dále pro manipulování s detekovanými variantami ve formátu VCF nebo BCF. [20][21]

[1] Z angl. quality score
[2] Z angl. variant calling

GATK - Genome Analysis Toolkit

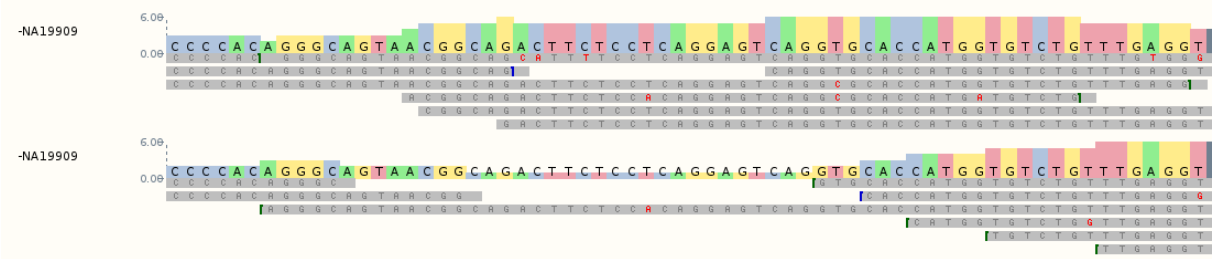
Průmyslový standart využívaný pro identifikaci jedno-nukleotidových polymorfismů (SNP) a indelů v DNA. Původně vyvinuté pro lidskou genetiku, ovšem nyní lze použít na genomová data z jakéhokoliv organismu.[22]

Freebayes

Freebayes je genetický detektor pro detekci variant, který slouží k nalezení malých polymorfismů. Mezi tyto polymorfismy řadíme SNP (jedno-nukleotidové polymorfismy), indely (inzerce a delece), MNP (multi-nukleotidové polymorfismy) a komplexní události (složené inzerce a substituce).[23]

TVC - Torrent Variant Caller

Plugin na serveru Ion Torrent, který volá jedno-nukleotidové polymorfismy (SNP), multi-nukleotidové polymorfismy (MNP) a indely ve vzorku přes odkaz nebo v cílené podmnožině této reference.[15]



Obrázek 4: Zobrazení referenčního genomu v porovnání se zarovnaným osekvenovaným genonem [24]

Pro úplnou rekapitulaci a upřesnění nejasností ještě zopakuji veškeré kroky nutné před detekcí variant. Nejprve je celý genom osekvenován do souboru, nejčastěji FASTQ formát. Z tohoto souboru se sekvence musí zarovnat s referenčním genomem a výstupem daného průběhu je soubor SAM nebo BAM. Následně se určuje, kde se zarovnané čtení liší od referenčního genomu a tato skutečnost je zapsána do VCF souboru. Tuto skutečnost můžeme vidět na obrázku 4, kde jsou změny označeny červeně.[15][19]

2.2.3 Terciární analýza

Poslední nejdůležitější krok analýzy se zabývá interpretací dat. Analýza se snaží najít spojení mezi variantními daty a fenotypem pozorovaným u pacienta. Terciární analýza začíná anotací variant, která poskytuje další informace. Po anotaci variant následuje filtrování dat, stanovení priorit a vizualizace dat.[15][19]

2.2.3.1 Anotace variant

Prvním klíčovým krokem pro analýzu je anotace variant. Jak již bylo dříve zmíněno, výstupem pro detekci variant je soubor ve formátu VCF, kde se nachází informace o variantách jako je pozice na genomu nebo referenční a alternativní zápis, ovšem nenajdeme zde žádné informace o biologických následcích. A právě nástroje, které provádějí analýzu variant se snaží dát těmto datům biologický kontext.[15][19]

2.2.3.2 Filtrování dat

Po dokončení anotace existuje stále velké množství variant. Pro tak velký počet není možné určit variantu, která způsobuje onemocnění, proto se používají nástroje na filtrování. Díky tisícům údajům jednotlivců v populační databázi je snazší rozpoznat některé nemoci. Jeden z nejčastějších filtrů se zaměřuje na výskyt alel (konkrétní forma genu) ve variantách (MAF).[15]

3 VCF formát

VCF formát je textový soubor, který se používá v bioinformatice pro uložení variant genových sekvencí. V daném souboru se nachází meta-informace, každý řádek s touto informací má na začátku předponu ##. Dále soubor obsahuje hlavičku, která má na začátku #. Následně již najdeme řádky s jednotlivými daty. Ke kódování znaků VCF souboru se využívá UTF-8.[25][26] Kompletní specifikaci nalezneme na následujícím odkazu: https://samtools.github.io/hts-specs/VCFv4.3.pdf.

```
##fileformat=VCFv4.3
##fileDate=20090805
##source=myImputationProgramV3.1
##reference=file:///seq/references/1000GenomesPilot-NCBI36.fasta
##contig=<ID=20,length=62435964,assembly=B36,md5=f126cdf8a6e0c7f379d618ff66beb2da,species="Homo sapiens",taxonomy=x>
##phasing=partial
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency">
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 129">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FILTER=<ID=q10,Description="Quality below 10">
##FILTER=<ID=s50,Description="Less than 50% of samples have data">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##FORMAT=<ID=HQ,Number=2,Type=Integer,Description="Haplotype Quality">
#CHROM POS     ID        REF    ALT     QUAL FILTER INFO                              FORMAT      NA00001        NA00002        NA00003
20     14370   rs6054257 G      A       29   PASS   NS=3;DP=14;AF=0.5;DB;H2           GT:GQ:DP:HQ 0|0:48:1:51,51 1|0:48:8:51,51 1/1:43:5:.,.
20     17330   .         T      A       3    q10    NS=3;DP=11;AF=0.017               GT:GQ:DP:HQ 0|0:49:3:58,50 0|1:3:5:65,3   0/0:41:3
20     1110696 rs6040355 A      G,T     67   PASS   NS=2;DP=10;AF=0.333,0.667;AA=T;DB GT:GQ:DP:HQ 1|2:21:6:23,27 2|1:2:0:18,2   2/2:35:4
20     1230237 .         T      .       47   PASS   NS=3;DP=13;AA=T                   GT:GQ:DP:HQ 0|0:54:7:56,60 0|0:48:4:51,51 0/0:61:2
20     1234567 microsat1 GTC    G,GTCT  50   PASS   NS=3;DP=9;AA=G                    GT:GQ:DP    0/1:35:4       0/2:17:2       1/1:40:3
```

Obrázek 5: Ukázka VCF souboru [25]

3.1 Datové typy

Ve VCF souborech je povoleno následujících 5 datových typů: Integer, Float, Flag(Double), Character a String.[25]

3.2 Meta-informace

Meta-informace jsou zobrazeny za předponou ## v páru klíč=hodnota. Tyto informace nutně nemusí být v souboru, jsou volitelné, ovšem doporučuje se zahrnutí těchto informací, jež jsou v těle VCF souboru. Ty řádky, kde klíč obsahuje párové tagy „<>“ je nutné zvolit ID, jež musí být v rámci svého typu jedinečné.[25][15] Zde je seznam typů meta-informací, jež se používají u VCF formátu:

- **fileformat** - obsahuje číslo verze VCF formátu
- **reference** - odkazuje na soubor s genomem
- **INFO** - popis vlastností, které najdeme ve sloupci INFO
- **FILTER** - filtry používané na data

- **FORMAT** - slouží pro popis genotypů
- **ALT** - symbolické alternativy pro nepřesné strukturální varianty
- **assembly** - určuje umístění souboru FASTA
- **contig** - značka popisující referenční sekvence v souboru
- **SAMPLE** - definuje mapování do genomu
- **PEDIGREE** - slouží k zaznamenávání vztahů mezi genomy nebo odkazuje na databázi

Jednotlivé meta-informace mohou být v jakémkoliv pořadí, ovšem „fileformat“ musí být vždy na první řádku.[25]

3.3 Hlavička a řádky s daty

Hlavička obsahuje 8 povinných položek:

- **CHROM** - identifikátor chromozómu
- **POS** - pozice varianty na chromozómu
- **ID** - identifikátor varianty
- **REF** - nukleotidový zápis referenční báze
- **ALT** - nukleotidový zápis alternativní báze
- **QUAL** - kvalita určující provedené měření v ALT
- **FILTER** - pole, které informuje, pokud je filtrace variant úspěšná
- **INFO** - pole, kde najdeme doplňující informace, které jsou obvykle popsány v hlavičce souboru

Pokud se v souboru nachází data o genotypech, jsou zobrazeny pod sloupcem **FORMAT**.[25][15] Jednotlivé sloupce jsou od sebe odděleny tabulátory. V případě neznámé hodnoty se píše tečka(.).

4 Další formáty pro práci s DNA

V bioinformatice se používá mnoho formátu pro práci s DNA. V této kapitole bych chtěl přiblížit aspoň pár z nich.

4.1 Formáty pro uložení genomu

Na přelomu tohoto tisíciletí byl kompletně osekvenován první lidský genom. Tyto informace bylo potřeba uložit a proto vznikly formáty, které se zaměřují na práci s genomem. Sekvenční formát definuje povolené rozložení a obsah textu v souboru.[27][28][29] Zde je ukázka formátů sekvencí.

- **Plain sequency format** - soubor v tomto formátu, může obsahovat pouze jednu sekvenci, oproti tomu většina jiných formátů obsahuje několik sekvencí v jednom souboru, v souboru se mohou vyskytovat pouze mezery a znaky IUPAC [27][28]

```
ACAAGATGCCATTGTCCCCCGGCCTCCTGCTGCTGCTGCTCTCCGGGGCCACGGCCACCGCTGCCCTGCC
CCTGGAGGGTGGCCCCACCGGCCGAGACAGCGAGCATATGCAGGAAGCGGCAGGAATAAGGAAAAGCAGC
CTCCTGACTTTCCTCGCTTGGTGGTTTGAGTGGACCTCCCAGGCCAGTGCCGGGCCCCTCATAGGAGAGG
AAGCTCGGGAGGTGGCCAGGCGGCAGGAAGGCGCACCCCCCCAGCAATCCGCGCGCCGGGACAGAATGCC
CTGCAGGAACTTCTTCTGGAAGACCTTCTCCTCCTGCAAATAAAACCTCACCCATGAATGCTCACGCAAG
TTTAATTACAGACCTGAA
```

Obrázek 6: Plain sequency format [28]

- **FASTQ format** - k ukládání biologické sekvence a jejího skóre kvality, soubor může obsahovat více sekvencí [27][28][30]
- **EMBL format** - soubor může obsahovat několik sekvencí, nýbrž začátek záznamu sekvence je vždy následující - první identifikační řádek, poté řádky s poznámkami a začátek sekvence je označen „SQ“ a konec dvěma lomítky „//“ [27][28]
- **FASTA format** - soubor ve kterém může býti více sekvencí, každá sekvence začíná jednořádkovým popisem začínající symbolem „>“, na následujícím řádku je již sekvence dat [27][28]

```
>AB000263 |acc=AB000263|descr=Homo sapiens mRNA for prepro cortistatin like peptide, complete cds.|len=368
ACAAGATGCCATTGTCCCCCGGCCTCCTGCTGCTGCTGCTCTCCGGGGCCACGGCCACCGCTGCCCTGCC
CCTGGAGGGTGGCCCCACCGGCCGAGACAGCGAGCATATGCAGGAAGCGGCAGGAATAAGGAAAAGCAGC
CTCCTGACTTTCCTCGCTTGGTGGTTTGAGTGGACCTCCCAGGCCAGTGCCGGGCCCCTCATAGGAGAGG
AAGCTCGGGAGGTGGCCAGGCGGCAGGAAGGCGCACCCCCCCAGCAATCCGCGCGCCGGGACAGAATGCC
CTGCAGGAACTTCTTCTGGAAGACCTTCTCCTCCTGCAAATAAAACCTCACCCATGAATGCTCACGCAAG
TTTAATTACAGACCTGAA
```

Obrázek 7: FASTA format [28]

- **GCG format** - soubor, kde je povolena pouze jedna sekvence, začíná řádkem s ID a následuje poznámky, sekvenční data začínají po řádku, jež má na konci dvě tečky „..“ [27][28]

- **GenBank format** - soubor, jež může obsahovat mnoho sekvencí, sekvence začíná klíčovým slovem „LOCUS“, dále jsou řádky s poznámkami, samotné sekvenční data začínají na dalším řádku za slovem „ORIGIN“ a končí dvěma lomítky „//“ [27][28]

4.2 Ostatní formáty

V předchozí podkapitole se objevily formáty, které pracují se prvotními daty neboli jsou to formáty pro uložení genomu. Nyní bych chtěl zmínit pár dalších formátu, na nichž můžeme narazit v bioinformatice.[27]

- **ASN.1** - strukturovaný formát využívaný v NCBI pro data DNA a proteinů
- **BCF** - komprimovaný formát VCF v binární podobě
- **BED** - formát, jež slouží pro popis genů a dalších vlastností sekvencí DNA
- **GFF** - používá k popisu genů a dalších funkcí DNA, RNA a proteinových sekvencí
- **GTF** - uchovává informace o genové struktuře
- **NEXUS** - strukturovaný blokový formát, který kóduje smíšené informace o datech genetické sekvence
- **NeXML** - formát XML používaný pro fylogenetický strom neboli „strom života“, který zobrazuje příbuzenské vztahy mezi různými biologickými druhy u nichž by měl být společný předek [31]
- **SAM** - textový formát používaný pro ukládání sekvenčních dat do sloupců oddělených tabulátory
- **BAM** - komprimovaný formát SAM v binární podobě

5 Implementace

Cílem mé práce bylo vytvořit program, který bude schopen zobrazit vybrané sekvence genomu a genetické mutace ve formátu VCF. Pro implementaci jsem se rozhodl využít programovací jazyk C#, který je používán při tvorbě aplikací .NET. Program je založen na architektuře .NET Framework.

5.1 Programovací jazyk C#

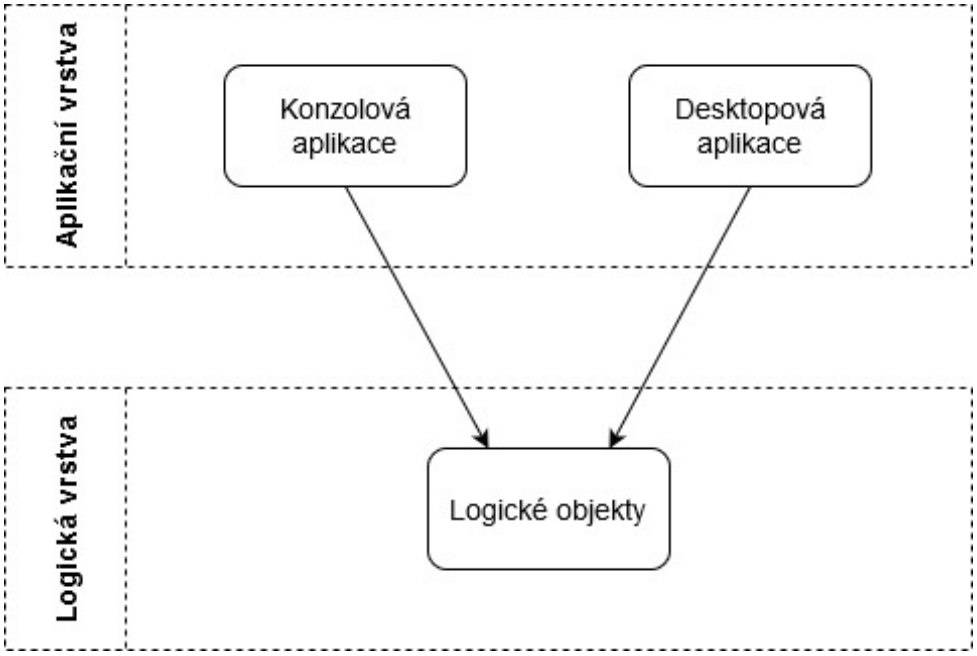
C# neboli též označovaný jako C-Sharp je objektově-orientovaný jazyk vytvořený společností Microsoft. Jeho syntaxe je podobná programovacím jazykům C++ nebo Java. První verze jazyka byla vydána v roce 2002.[32]

5.2 Architektura .NET

Architektura Microsoft .NET je programovací model pro platformu .NET. .NET Framework poskytuje spravované prováděcí prostředí, zjednodušený vývoj a integraci s celou řadou programovacích jazyků. Knihovna tříd .NET Framework je komplexní objektově orientovaná kolekce opakovaně použitelných typů, které můžete použít k vývoji aplikací.[33]

5.3 Rozložení systému

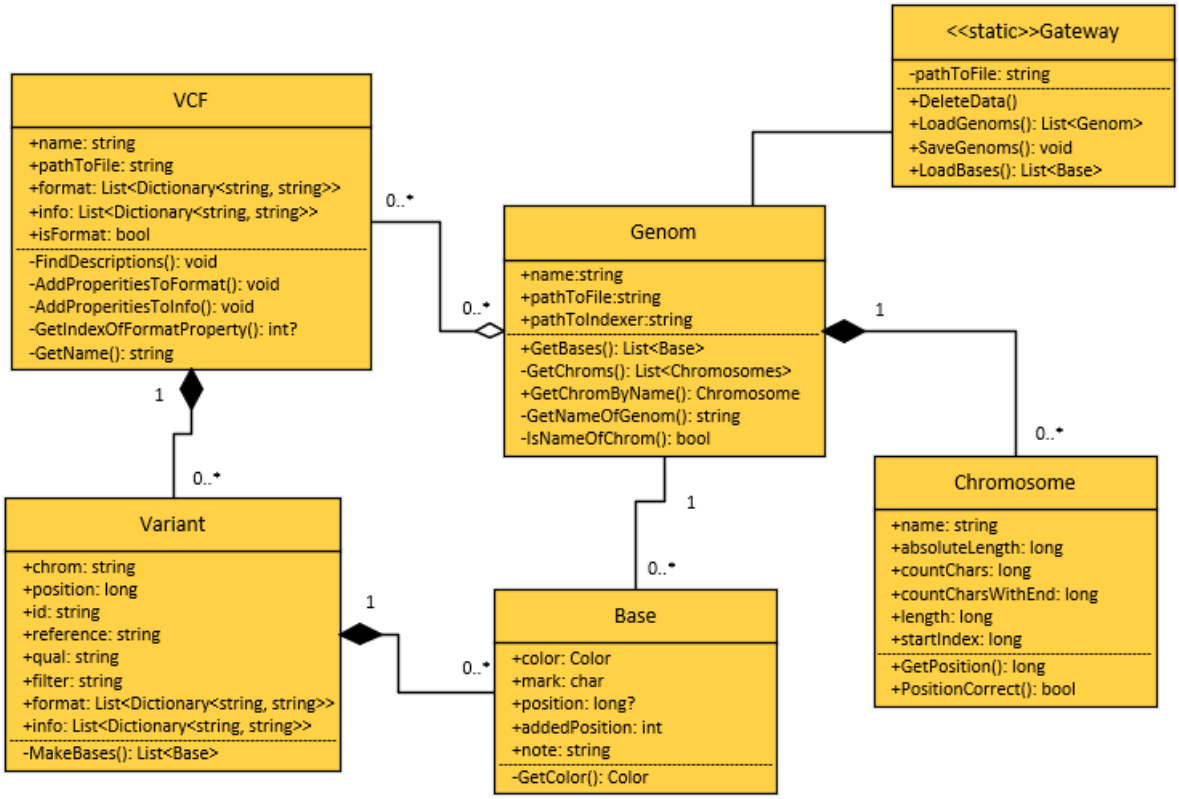
V této kapitole se věnuji rozložení systému a důvodům proč jsem zvolil tohle řešení. V první fázi jsem pracoval pouze s konzolovou aplikací. Ta mi poskytovala rychlou a snadnou přípravu funkcí, které byli potřeba k vytvoření základního chodu programu. Program pracuje pouze se soubory, které jsou uloženy v počítači uživatele a práce s nimi není náročná, proto nebylo třeba použít datovou vrstvu. Z těchto důvodu je systém rozložen na vrstvy aplikační a logickou, což můžeme vidět na obrázku 8. V samotném programu konzolovou aplikaci nenajdeme, sloužila pouze pro vývoj.



Obrázek 8: Rozložení systému

5.4 Diagram tříd

Aplikace je vyvíjena objektově, tudíž ji je nutné tak navrhnout. K popisu je využit diagram tříd, který můžeme vidět na obrázku 9. Třída `Genom` je souborem sekvencí nukleonových bází (třída `Base`). `Genom` se dělí na menší části tzv. chromozomy (třída `Chromosome`), což v diagramu reprezentuje kompoziční vztah. Aplikace pracuje s tím, že každému genomu můžou být přiřazeny VCF soubory, kvůli tomu je použitý vztah agregace. VCF soubor obsahuje různé varianty genových mutací(třída `Variant`). K reprezentaci jednotlivých bází, jak u genomu tak i genových mutací, slouží třída `Base`, která v sobě uchovává znak báze (`mark`), pozici (`position`) a barvu (`color`). Uživatelský komfort zvyšuje třída `Gateway`, která je určena pro ukládání a načítání genomů a VCF souborů, se kterými uživatel pracuje.



Obrázek 9: Třídní diagram

5.5 Popis implementace

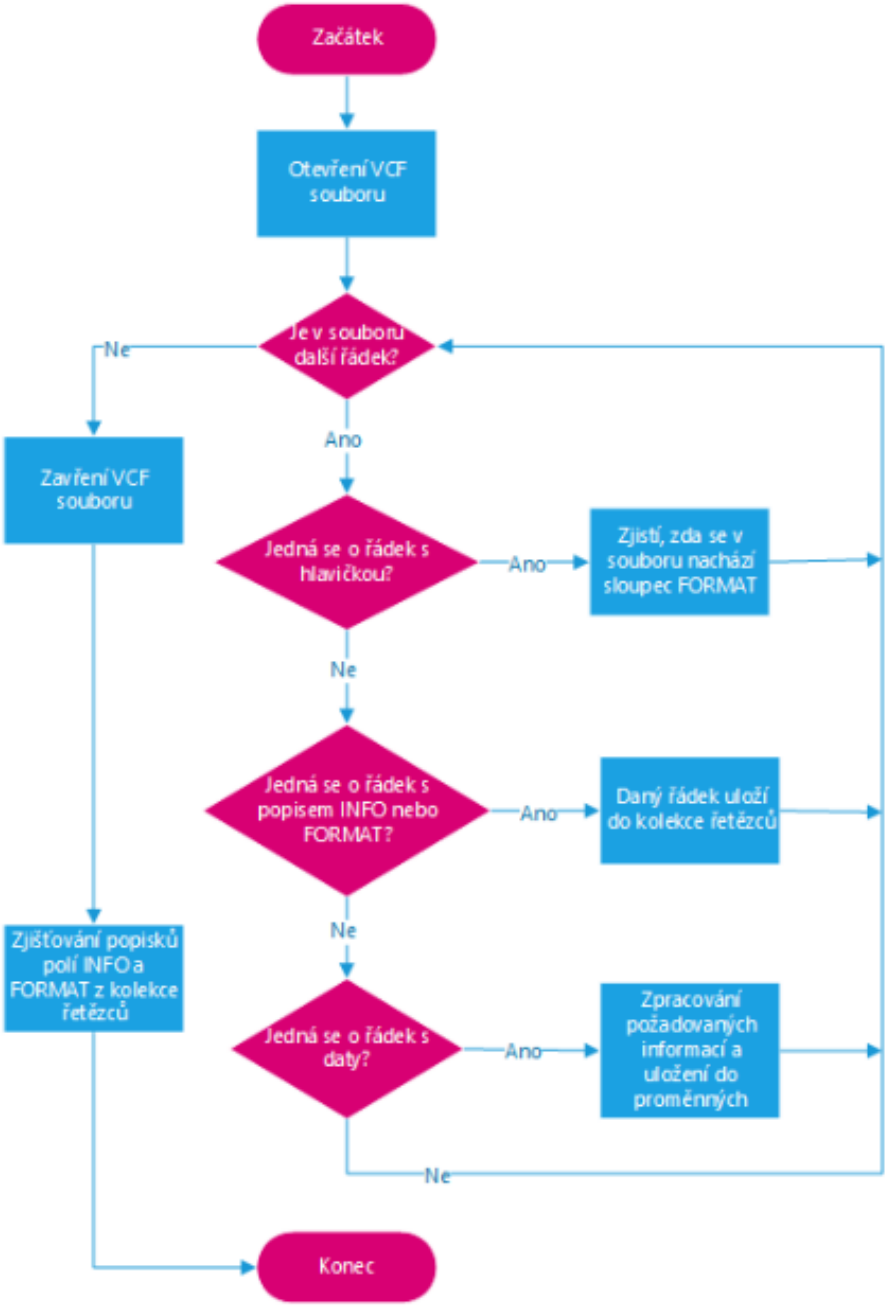
Aplikace má různé funkce, nyní bych chtěl přiblížit, jak jsou tyto funkce implementovány. Každá podkapitola obsahuje diagram, který zjednodušeně zobrazuje daný proces.

5.5.1 Načtení VCF souborů

Jedním z hlavních úkolů je vizualizace VCF souboru. Abychom soubor mohli zobrazit, je nutné nejdříve zpracovat data, která obsahuje. Postup při práci se souborem můžeme vidět na obrázku 10. Jak diagram ukazuje, soubor je načítán po řádku.

Aplikace dokáže zobrazit časovou řadu vývoje, kde jsou potřeba údaje z nepovinného sloupce FORMAT, pokud soubor nemá tento sloupec, není v nabídce pro zobrazení v grafu. Všechny řádky, které slouží pro popis vlastností ve sloupcích INFO a FORMAT, nejdříve musím uložit do kolekce, jelikož VCF soubory nemusí obsahovat popis všech vlastností nebo naopak obsahují i některé, které nejsou součástí variant, tudíž se této problematice věnuji až posléze. Pokud se jedná o samotný řádek s variantou, řádek se rozdělí na samotné sloupce a poté z něj vytvářím objekt třídy `Variant`. Některá z variant může obsahovat více alternativních zápisů na jeden referenční zápis, což můžeme vidět na obrázku 5. Na 3. řádku se samotnými varianty můžeme

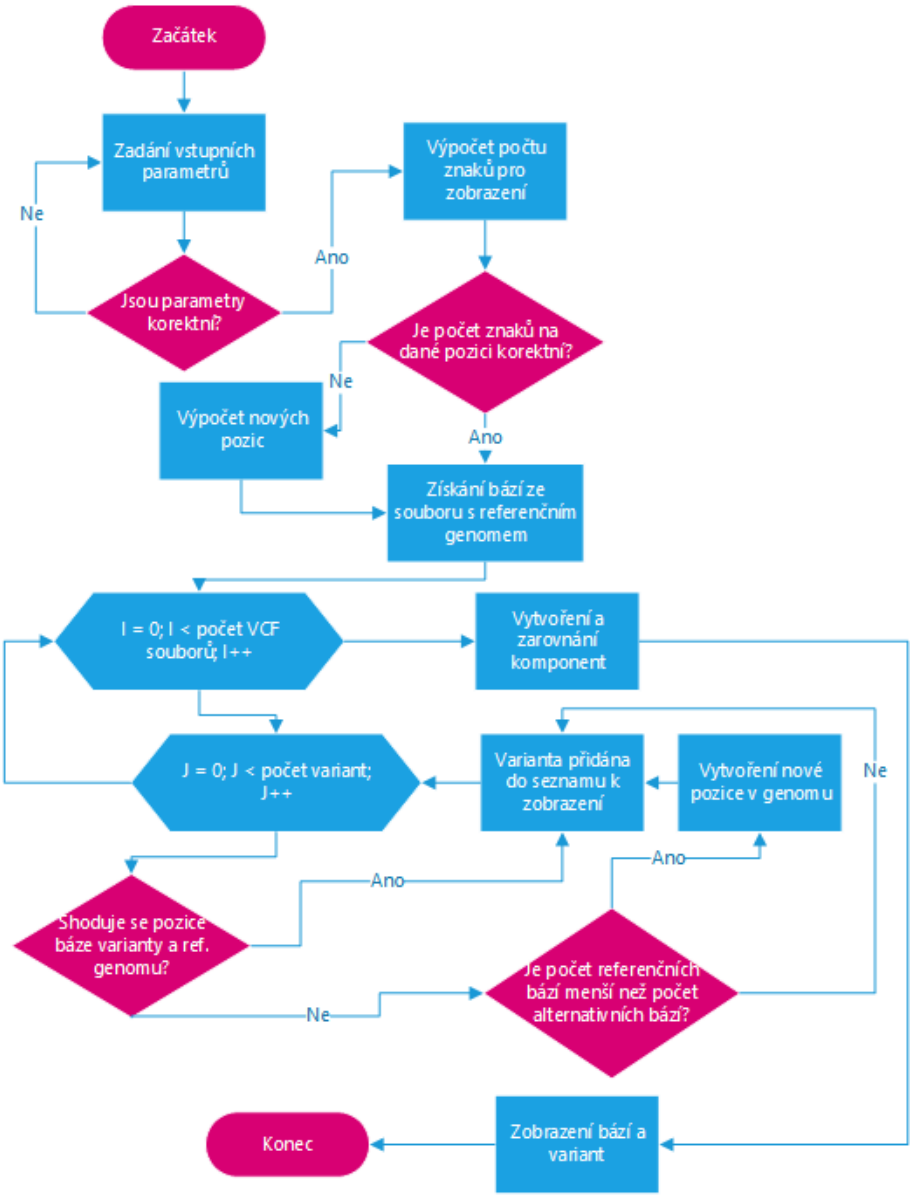
vidět referenční zápis A a 2 alternativní zápisy G,T. Aplikace vyřeší potíž vytvořením varianty pro každý alternativní zápis, tj. z jednoho řádku se vytvoří dvě varianty.



Obrázek 10: Diagram načtení VCF souboru

5.5.2 Zobrazení sekvence lidského genomu a genetické mutace

Na diagramu 11 je vidět zjednodušený proces, který probíhá při zobrazování sekvencí lidského genomu a genových mutací. Podrobnějšímu řešení se věnuji v další části, kde vysvětluji různé výpočty a důvody, proč jsem postupoval daným způsobem.



Obrázek 11: Diagram zobrazení sekvencí lidského genomu a genových mutací

Pro zobrazení bází jsem použil komponentu `Panel`, která slouží jako kontejner pro vkládání dalších komponent. Do panelu jsou následně vkládány komponenty `Label`. `Label` je ovládací prvek, pomocí nějž můžeme zobrazit text nebo obrázky. V mém případě jedna komponenta `Label` reprezentuje jeden objekt třídy `Base`. Vstupními údaji jsou genom, chromozom a pozice

na chromozomu, následně je zavolána funkce `ShowBases()`, která v první fázi ověří vstupní údaje. Pokud jsou údaje v pořádku, funkce pokračuje druhou části, kde je potřeba vypočítat počet znaků, které se svou velikostí vlezou do panelu. Uživatel si může tento počet v omezené míře měnit, což vidíme na obrázku 16, kde je tato vlastnost zobrazena jako procentuální číslo. V podstatě tohle číslo definuje mezeru mezi jednotlivými znaky.

Pro samotný výpočet je zapotřebí znát šířku komponenty `Panel`, šířku komponenty `Label` a velikost mezery mezi jednotlivými znaky. Na výpisu 1 můžeme vidět samotný výpočet. Nejprve je spočítán celkový počet znaků, který se vleze na danou šířku panelu. Pokud je počet znaků sudý, tak k celkovému počtu přičteme další znak, protože pozice komponent `Label` je počítána od středu panelu. Abychom dostali efekt zobrazení pouze části znaku, je nutno přičíst ještě dva znaky.

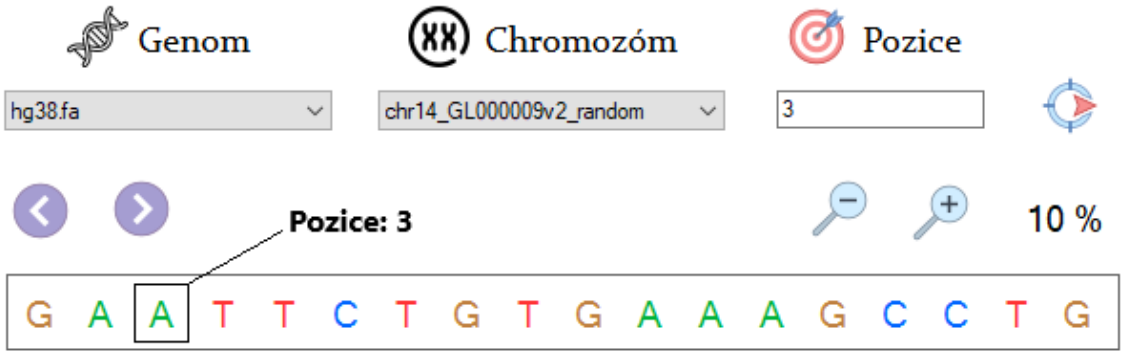
```
int countChars = characterPanel.Width / (labelWidth + spaceWidth);

if (countChars % 2 == 0) countChars++;
countChars += 2;

int countCharsForEachSide = (countChars - 1) / 2;
```

Výpis 1: Výpočet počtu bází pro zobrazení

Funkce, která se stará o načtení jednotlivých bází ze souboru, pracuje s počtem znaků, které se mají zobrazit na každou stranu od dané pozice. Představme si, že jsme zadali pozici na chromozomu např. 12000 a celkový možný počet znaků k zobrazení je např. 15. To znamená, že by se zobrazily báze na pozicích 11993 - 12007.



Obrázek 12: Zobrazení bází na prvních pozicích

Jestliže chceme načíst báze na prvních nebo posledních pozicích nastává problém, ten spočívá v nedostatečném počtu načtených bází. Například máme zadanou pozici 3, počet zobrazovaných bází je 19, tudíž na každou stranu od pozice by mělo být zobrazeno 9 znaků. Zde nastávají potíže, jelikož funkce není schopna vrátit požadovaný počet znaků a při získávání znaků by

se dostala do mínusového indexu. Pro vyřešení této záležitosti je potřeba upravit počáteční a koncovou pozici. Po upravení pozic, nám funkce vrátí 2 znaky před a 16 znaků po zadané pozici (16. znak není na obrázku viditelný, kvůli zarovnání znaků doleva). Jelikož se velikost souboru s genomem pohybuje v GB, tak při získávání bází aplikace nenačítá soubor s genomem řádek po řádku. Soubor se pouze programově otevře a pomocí funkce se přesune na vybranou pozici.

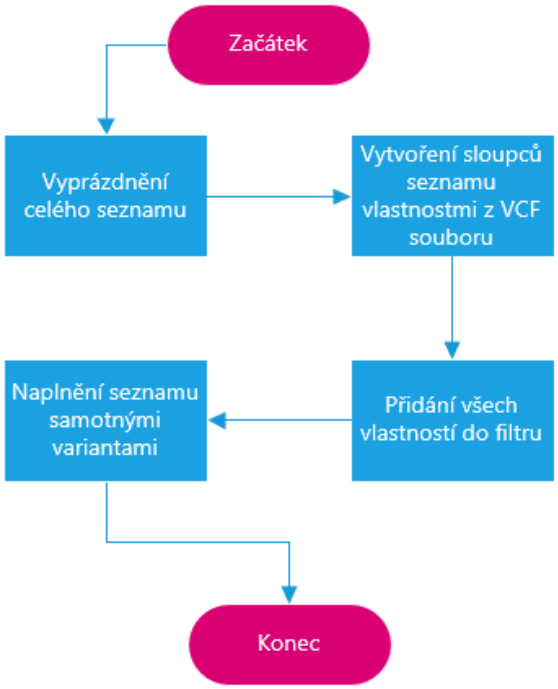
Nyní aplikace porovná varianty s načtenými bázemi. Každá varianta je procházena po jednotlivých znacích. Pokud se pozice znaku varianty shoduje s pozicí znaku genomu je znak varianty přidán do seznamu variant k zobrazení. Jestliže je znak varianty navíc neboli počet referenčních znaků je menší než počet alternativních znaků, tak je nutno přidat tento znak i do seznamu bází genomu. Tento krok zapříčiní větší počet znaků genomu než bylo vypočítáno, tudíž je potřeba odstranit od obou konců seznamu bází genomu nadbytečný počet znaků.

Po této komplikované části, následuje samotné zobrazení. Tady se aplikace chová podobně, jak jsem již vysvětloval v předchozích odstavcích, tj. pokud se zadaná pozice nachází na začátku chromozomu, tak jsou znaky zarovnány doleva, naopak pokud je pozice na konci chromozomu, aplikace zarovná znaky doprava. Jestliže se pozice nachází někde uprostřed chromozomu, znaky se zarovnají středem, jak je vidět na obrázku 16.

Vizuální stránku genových mutací zobrazuji posléze. Proces začíná porovnáváním variant mutací s bázemi lidského genomu, které zbyli po ořezání. Jestliže se shoduje pozice varianty s pozicí báze, následuje zobrazení znaku. Pokud referenční zápis odkazuje na větší počet bází než alternativní zápis, jsou tyto znaky reprezentovány pomlčkou. Tuto skutečnost můžeme vidět na obrázku 16.

5.5.3 Zobrazení seznamu jednotlivých variant a filtrování vlastností

Varianty se zobrazují do komponenty `ListView`. Ještě před samotným vkládáním variant do seznamu, je nutné seznam vyprázdnit od předchozích variant. Dále je potřeba vytvořit požadovaný počet sloupců seznamu. Seznam všech sloupců je vytvářen již při načítání VCF souboru. Každý z prvních 7 údajů z VCF souboru (CHROM, POS, ID, REF, ALT, QUAL, FILTER) tvoří jeden sloupec, který má nastavenou defaultní šířku. Každá doplňující vlastnost, které spadá pod hlavičku INFO, reprezentuje jeden sloupec v seznamu. To samé platí, i u sloupce FORMAT. Tyto sloupce nejsou zprvu v seznamu viditelné, jelikož jejich šířka je nastavena na 0. Souběžně s vytvářením sloupců probíhá i vytvoření filtru, kde se nachází vlastnosti sloupců INFO a FORMAT.



Obrázek 13: Diagram zobrazení variant s vlastnostmi

Po vytvoření sloupců následuje vložení variant do seznamu. Tady nastává potíž, jelikož každá varianta, může obsahovat různý počet vlastností, myšlen hlavně sloupec INFO. Nutno podotknout, že objekt třídy `VCF` obsahuje celkový seznam vlastností, které se ve VCF souboru vyskytují, oproti tomu objekt třídy `Variant` obsahuje pouze ty údaje, které obsahuje jedna daná varianta. Při vkládání nejprve procházím jednotlivé varianty a ke každé vlastnosti hledám index vytvořeného sloupce, do kterého je následně údaj vložen.

Při vytváření programu jsem zprvu vkládal údaje přímo do komponenty `ListView`, ovšem to zapříčinilo zdlouhavé zobrazovaní. Pokud by VCF soubor obsahoval stovky variant, mohlo by načítání trvat i pár desítek vteřin. Proto jsou údaje nejprve vkládány do kolekce a poté je celá kolekce teprve vložena do komponenty `LitView`.

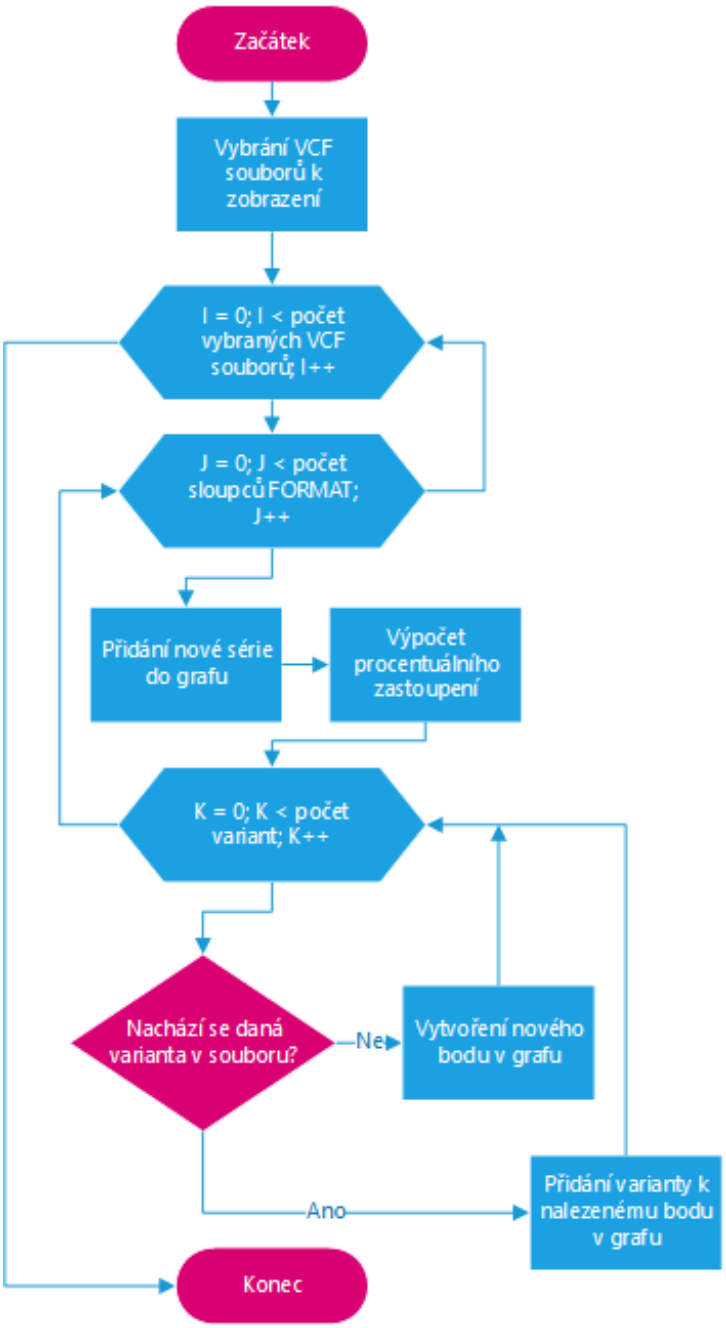
Jak jsem již výše zmínil, při prvním zobrazení je vidět pouze prvních 7 údajů. Další jednotlivé vlastnosti si může uživatel zobrazit zaškrtnutím políčka s příslušným názvem, tj. po každé změně, ať už zaškrtnutí nebo odškrtnutí, se porovnají sloupce ze seznamu se zaškrtnutými vlastnostmi z filtru. Pokud daný sloupec není nalezen v zaškrtnutých položkách, je jeho šířka nastavena na 0. Jestliže dojde ke shodě sloupce s položkou z filtru a zároveň šířka sloupce se rovná 0, tak dojde k nastavení šířky sloupce na defaultní hodnotu.

5.5.4 Vizualizace časové řady vývoje

Časová řada vývoje se zobrazuje do komponenty `Chart`. Tato komponenta slouží k zobrazování grafu, v našem případě se používá k zachycení vývoje konkrétní mutace u jednoho jedince v čase

nebo pro vizualizaci dvou vzorků jednoho jedince pocházejících z různých tělních tkání. Obecný diagram zobrazení časové řady je vylíčen na obrázku 14. Proces začíná vytvořením seznamu VCF souborů, které obsahují sloupec FORMAT. Následuje zobrazení seznamu s těmito vyfiltrovanými VCF soubory, kde si uživatel vybere ty soubory, které chce zobrazit do grafu. Poté se vybrané VCF soubory prochází v cyklu, kde každý sloupec FORMAT je taktéž procházen (na obrázku 5 sloupce NA00001, NA00002, ...). Při každém opakování se do grafu přidává nová série. Jedna série představuje jeden sloupec ve VCF souboru, v případě, že soubor obsahuje pouze jeden sloupec, tak reprezentuje celý VCF soubor, příklad na obrázku 20, kde legenda zobrazuje 2 VCF soubory, ovšem první soubor obsahuje již zmíněný větší počet sloupců FORMAT.

Nyní dochází k porovnání variant. Každá série je jednotlivě procházena a u každé se porovnávají její vytvořené varianty (jednotlivé body na ose x) a ty které mají shodné porovnávací parametry (chromozóm, pozice, reference, alternativa) se zde přidají. V opačném případě, kdy se varianta ještě nenachází v grafu, se nakonec osy x přidá další bod, na který je přidána varianta.



Obrázek 14: Diagram zobrazení časové řady vývoje

6 Experimentální ověření programu

Abychom ověřili, zda skutečně aplikace pracuje správně, ověříme ji jednoduchými experimenty. Všechny VCF soubory, které budu používat k testům, jsou dostupné v příloze.

6.1 Zobrazení genových mutací filtrování vlastností

Abych byl schopen demonstrovat, zda aplikace zobrazí báze referenčního genomu nebo variant, vytvořil jsem VCF soubor, kde najdeme možnosti, které můžou při zobrazování nastat. Na obrázku 15 najdeme daný soubor, který obsahuje jak meta-data, tak i samotné varianty (řádky, které nezačínají znakem #). Sloupce FORMAT jsem zakomentoval, jelikož nejsou důležitou součástí textu a obrázek by nebyl čitelný.

```
##fileformat=VCFv4.2
##FORMAT=<ID=AD,Number=.,Type=Integer,Description="Allelic depths for the ref and alt alleles in the order listed">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Approximate read depth (reads with MQ=255 or with bad mates are filtered)">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=PL,Number=G,Type=Integer,Description="Normalized, Phred-scaled likelihoods for genotypes as defined in the VCF specification">
##INFO=<ID=AC,Number=A,Type=Integer,Description="Allele count in genotypes, for each ALT allele, in the same order as listed">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency, for each ALT allele, in the same order as listed">
##INFO=<ID=AN,Number=1,Type=Integer,Description="Total number of alleles in called genotypes">
##INFO=<ID=BaseQRankSum,Number=1,Type=Float,Description="Z-score from Wilcoxon rank sum test of Alt Vs. Ref base qualities">
##INFO=<ID=STR,Number=0,Type=Flag,Description="Variant is a short tandem repeat">
#CHROM  POS ID  REF ALT QUAL    FILTER  INFO    FORMAT  NA00001 NA00002
chr12   133275309   .   G   GGG 256 .   AC=1;AF=0.500;AN=5  GT:AD:DP:GQ:PL  ... ...
chr17   7668836 .   GAA GAT,G   92.73   .   AC=1,1;AN=3;STR GT:AD:DP:GQ:PL  ... ...
chr17   7668880 .   C   GTAT    42992.77    .   AC=1;AF=0.500;AN=2  GT:AD:DP:GQ:PL  ... ...
chr17   7668838 rs1625895   T   C   150990.77   .   AC=2;AF=1.00;AN=8   GT:AD:DP:GQ:PL  ... ...
```

Obrázek 15: Ukázka VCF souboru 2019-12-14.vcf

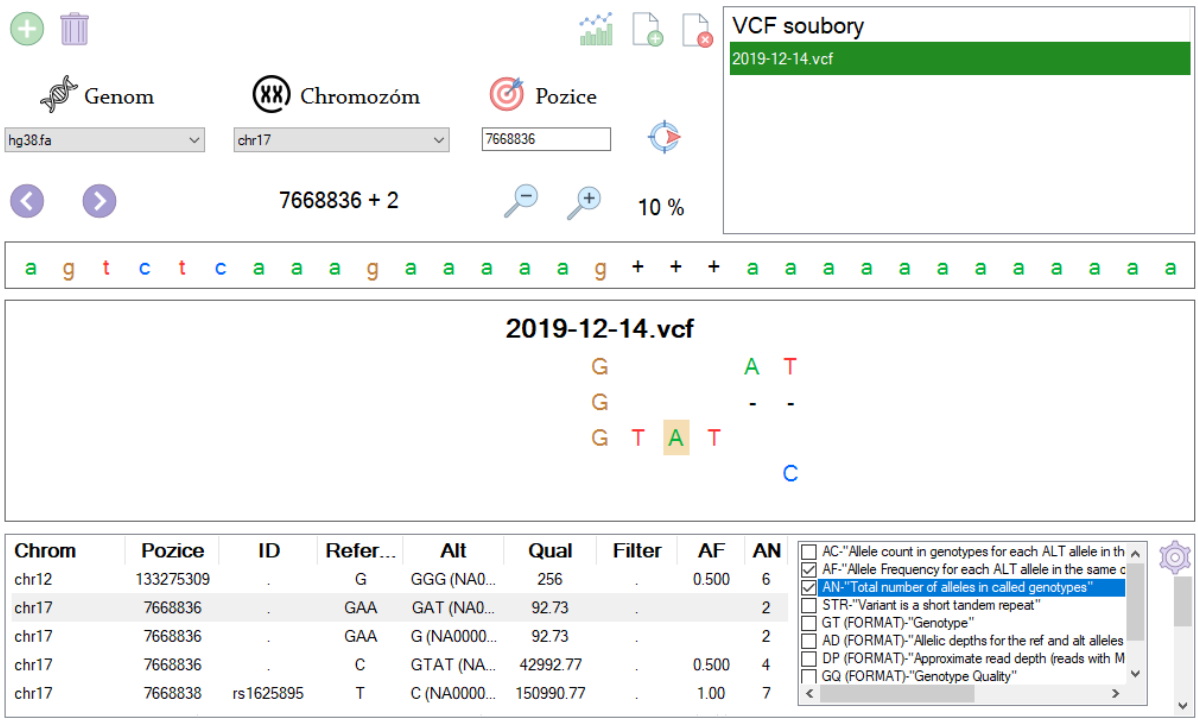
My se budeme nyní zabývat pouze variantami. Když se podíváme na obrázek 16, tak v panelu pro zobrazení variant (prostřední panel, kde je tučným písmem napsán název VCF souboru 2019-12-14.vcf) jsou zobrazeny 4 varianty. To by odpovídalo i počtu řádků ve VCF souboru, ovšem již první řádek s daty se nachází na jiném chromozomu, tudíž ho nyní nelze vidět. Jak jsem psal již v kapitole 5.5.1, pokud řádek obsahuje více alternativních zápisů, aplikace to bude chápat jako dvě varianty. A tato situace nastane na druhém řádku s variantami v souboru.

Při zobrazování variant můžou nastat tři věci, buď se může jednat o substituci, kde se počet znaků alternativního zápisu rovná počtu znaků referenčního zápisu. Tento případ v souboru najdeme na druhém a čtvrtém řádku. A jak je vidno, tak se substituce i správně promítla do panelu. V první zobrazené variantě je alternativní zápis GAT zobrazen na stejných pozicích jako referenční zápis GAA, to samé platí i pro poslední zobrazenou variantu, kde alternativa C se zobrazuje pod referencí T.

Další případ, který může nastat, nazýváme delece. Nastává, pokud referenční zápis obsahuje větší počet bází než alternativa. Tento jev se v souboru nachází na prvním a druhém řádku. Na obrázku 11 můžeme vidět, že varianta z druhého řádku se i v panelu pro varianty správně zobrazila. Chybějící znaky jsou nahrazeny pomlčkou, tudíž alternativní zápis není G, ale G - -.

Poslední možností je inserce (adice), která nastává pokud alternativní zápis čítá více znaků než reference, tudíž je potřeba přidat přídavné báze do referenčního genomu, tam ho reprezen-

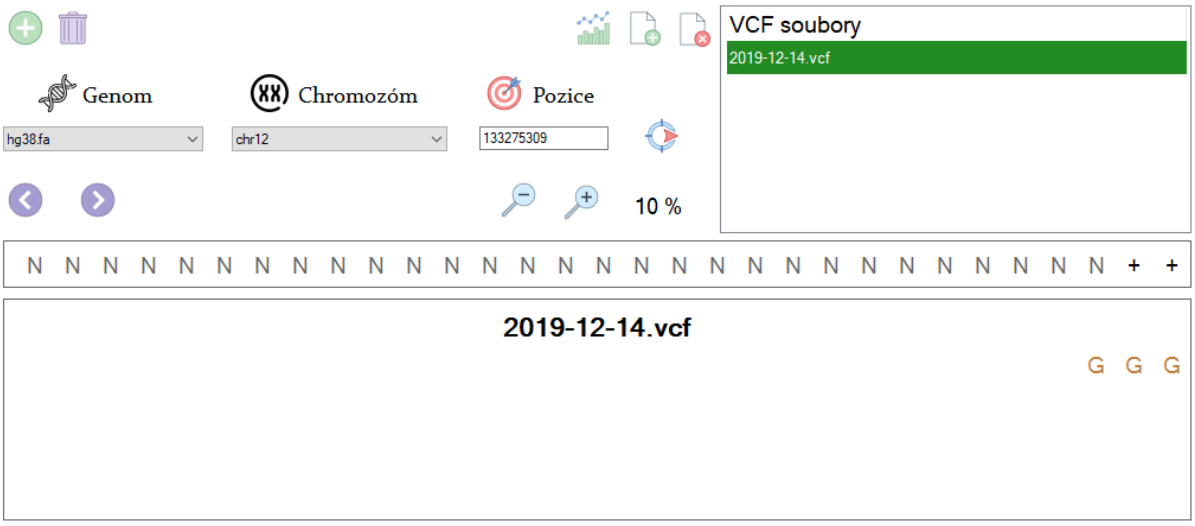
tuje znak +. V souboru se tato varianta nachází na třetím řádku s variantami. Pod výběrem chromozómu můžeme vidět pozici báze, v tomhle případě zobrazuje pozici znaku A, a protože se jedná o inserci, tak obsahuje i doplňující informaci o kolikátý znak navíc se jedná. Znak je zvýrazněn béžovou barvou. Aplikace tuto skutečnost zobrazila, jak bylo předpokládáno.



Obrázek 16: Hlavní okno aplikace pro zobrazení genomu a genových mutací

V panelu, který se nachází úplně dole můžeme vidět všechny varianty z vybraného souboru. Tento panel defaultně zobrazuje prvních 7 sloupců, ostatní vlastnosti si uživatel může vybrat v seznamu, který vidíme vpravo dole. Pokud porovnáme data z VCF souboru a údaje ze seznamu, tak je jasné, že program, tyto údaje vylíčil korektně.

Pro úplný test, jsem se rozhodl vyzkoušet, jestli se varianta správně zobrazí i na konci sekvence chromozómu. Variantu najdeme v souboru na prvním řádku s daty. Na obrázku 17 lze vidět, že varianta se vylíčila korektním způsobem.



Obrázek 17: Zobrazení inserce na konci chromozómu

6.2 Vizualizace časové řady

K otestování jsem vytvořil dva odlišné VCF soubory, jejich data s variantami můžeme vidět na obrázku 18 a obrázku 19. Tyto data jsou smyšlené a mají pouze demonstrovat funkčnost. Sloupec INFO jsem na obrázcích schoval, jelikož nebude v této části experimentu důležitý.

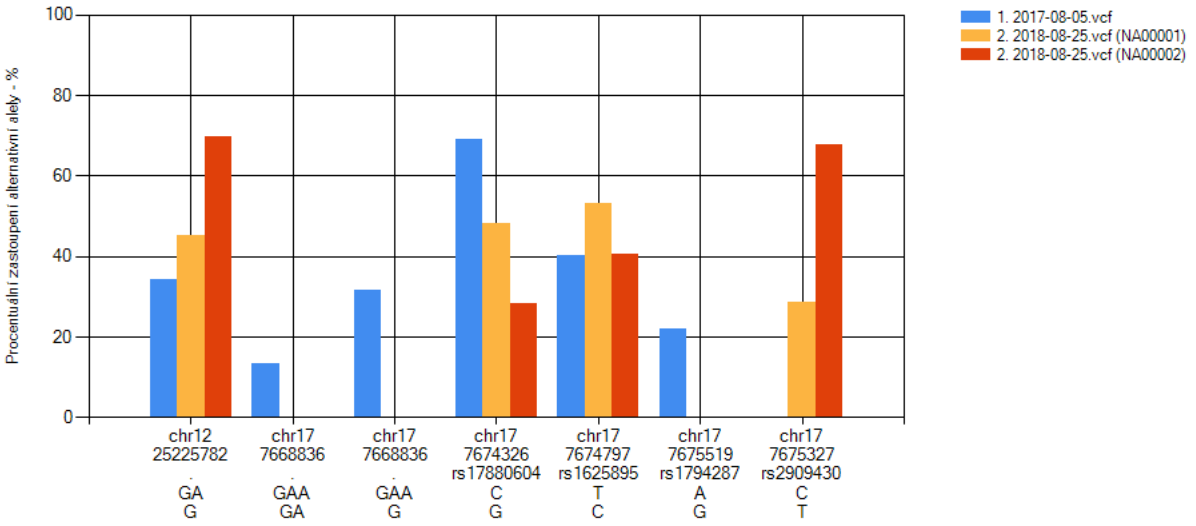
```
#CHROM  POS ID  REF ALT QUAL    FILTER  INFO    FORMAT  4695-JK-16_S82_L001_R12_001.primers_out
chr12   25225782    .   GA  G   4295.73 .   ... GT:AD:DP:GQ:PL  0/1:3487,1812:5299:99:4333,0,42262
chr17   7668836 .   GAA GA,G    91362.73    .   ... GT:AD:DP:GQ:PL  1/2:2833,689,1617:5139:99:91400,34375,37190,10460,0,34829
chr17   7674326 rs17880604  C   G   42992.77    .   ... GT:AD:DP:GQ:PL  0/1:881,1963:2844:99:43021,0,42255
chr17   7674797 rs1625895   T   C   150990.77   .   ... GT:AD:DP:GQ:PL  1/1:4020,2701:6721:99:151019,10721,0
chr17   7675519 rs1794287   A   G   191851.77   .   ... GT:AD:DP:GQ:PL  1/1:3821,1086:4907:99:191880,13954,0
```

Obrázek 18: Varianty ve VCF souboru 2017-08-05

```
#CHROM  POS ID  REF ALT QUAL    FILTER  INFO    FORMAT  NA00001 NA00002
chr12   25225782    .   GA  G   4295.73 .   ... GT:AD:DP:GQ:PL  0/1:4014,3294:7308:99:4333,0,42262  0/1:508,1167:1675:99:4333,0,42262
chr17   7674326 rs17880604  C   G   42992.77    .   ... GT:AD:DP:GQ:PL  0/1:2364,2189:4553:99:43021,0,42255 0/1:2998,1176:4174:99:43021,0,42255
chr17   7674797 rs1625895   T   C   150990.77   .   ... GT:AD:DP:GQ:PL  1/1:2777,3169:5946:99:151019,10721,0    1/1:3624,2485:6109:99:151019,10721,0
chr17   7675327 rs2909430   C   T   62537.77    .   ... GT:AD:DP:GQ:PL  1/1:1602,640:2242:99:62566,4264,0   1/1:1206,2541:3747:99:62566,4264,0
```

Obrázek 19: Varianty ve VCF souboru 2018-08-25

Na obrázku 20 se vykreslují časové řady vývoje. Když se podíváme na graf, tak vidíme, že soubory mají společné 3 varianty. Pokud si to ověříme v samotných souborech, tak první řádky s variantou si odpovídají, dále třetí a druhý řádek a poslední stejnou variantu najdeme na čtvrtém a třetím řádku.



Obrázek 20: Zobrazení časové řady vývoje v grafu

První VCF soubor, který je na obrázku 18, obsahuje pouze 5 řádků s variantami, ovšem v grafu (modré sloupce) máme vyobrazených 6 sloupců, tento jev nastává, pokud obsahuje některá z variant více alternativních zápisů, což druhá varianta v souboru splňuje. Nyní se pozastavíme u druhého souboru, jelikož soubor obsahuje více sloupců FORMAT. V těchto případech aplikace zobrazí dané údaje do grafu pro každý sloupec zvlášť.

Závěr

Při vytváření této práce jsem se dozvěděl mnoho informací o DNA, genetice a věcí s tímto spjatými. Seznámil jsem se se sekvenací genomu, což pro mě bylo dosud velkým neznámým. Vůbec celkově téma VCF souborů a jejich variant je velmi zajímavé, jelikož odhaluje odlišnosti v DNA a tím pádem např. i nemoci.

Vyvinul jsem aplikaci, která dokáže zobrazit genom ve formátu FASTA a následně oproti němu zobrazit genetické mutace ve formátu VCF. Při implementaci jsem narazil na hodně problémů. Mezi tyto problémy mohu zařadit kupříkladu zobrazení inserce, která byla poměrně velkým oříškem, jelikož mi aplikace po implementování nezobrazovala to, co by měla. Promítnutí vývoje do časové řady nebylo obtížné, ovšem nastával problém, pokud soubor obsahoval hodně variant. V grafu nešly přečíst jednotlivé popisky. Problém byl vyřešen přiblížením, kdy se zobrazí pouze vybraná část časových řad.

Postupný vývoj mi umožnil aplikaci odladit, kdy se při zobrazení variant program již nezasekne, jako to dělal v prvopočátku. V poslední kapitole se jasně ukazuje, že program dané sekvence zobrazuje správně.

V práci se najde určitě místo na vylepšení. Například aplikace by mohla umožnit uživateli použít i jiné vstupní soubory s genomem (FASTQ, EMBL, GCG) nebo binární verzi formátu VCF, tedy formát BCF.

Literatura

1. WIKIPEDIE. *Lidský genom — Wikipedie: Otevřená encyklopedie*. 2019. Dostupné také z: https://cs.wikipedia.org/w/index.php?title=Lidsk%C3%BD_genom&oldid=17419439. [Online; navštíveno 3. 05. 2020].
2. HANZELKOVÁ, Zuzana; LÍZAL, Pavel; JARKOVSKÁ, Michaela; MATONOHOVÁ, Martina; PASZEKOVÁ, Helena; SITTOVÁ, Martina; TEJKALOVÁ, Kateřina; VALOVÁ, Simona. *Od počátku s DNA: Multimediální elektronický výukový materiál* [online]. Brno: Masarykova univerzita, 2013 [cit. 2020-05-04]. Dostupné z: https://is.muni.cz/do/sci/UEBBiol/DNA-FTBcz/pages/3-39-genom.html.
3. DVOŘÁKOVÁ, Lenka. *Rozluštění lidského genomu* [online]. Akademon, 2001 [cit. 2020-05-04]. Dostupné z: http://www.akademon.cz/article.asp?source=genom.
4. *Genom* [online]. WikiSkripta, 2018 [cit. 2020-05-04]. Dostupné z: https://www.wikiskripta.eu/w/Genom.
5. WIKIPEDIE. *DNA — Wikipedie: Otevřená encyklopedie*. 2020. Dostupné také z: https://cs.wikipedia.org/w/index.php?title=DNA&oldid=18217106. [Online; navštíveno 4. 05. 2020].
6. *DNA (nukleová kyselina)* [online]. WikiSkripta, 2018 [cit. 2020-05-04]. Dostupné z: https://www.wikiskripta.eu/w/DNA_(nukleov%5C%C3%5C%A1_kyselina).
7. CVEK, Boris. *DNA a lidská identita* [online]. Praha: Britské listy, 2018 [cit. 2020-05-04]. Dostupné z: https://img.blisty.cz/img/-12527.jpg?id=-12527%5C&size=450%5C&mg=0.
8. *Chromozom* [online]. WikiSkripta, 2016 [cit. 2020-05-04]. Dostupné z: https://www.wikiskripta.eu/w/Chromozom.
9. ŠÍPEK, Antonín. *Chromozomy* [online]. Genetika - biologie, 2014 [cit. 2020-05-04]. Dostupné z: http://www.genetika-biologie.cz/chromozomy.
10. ŠÍPEK, Antonín. *Varianty lidského chromozomu 9 a jejich souvislosti* [online]. Gate2BioTech, 2012 [cit. 2020-05-04]. Dostupné z: http://www.gate2biotech.cz/varianty-lidskeho-chromozomu-a-jejich-souvislosti/.
11. *Cytogenetika* [online]. Brno: Veterinární a farmaceutická univerzita Brno, 2013 [cit. 2020-05-04]. Dostupné z: https://cit.vfu.cz/opvk2014/texty_cz/obr55.png.
12. ŠÍPEK, Antonín. *Genotyp* [online]. Genetika - biologie, 2014 [cit. 2020-05-10]. Dostupné z: http://www.genetika-biologie.cz/genotyp.
13. WIKIPEDIE. *Genotyp — Wikipedie: Otevřená encyklopedie*. 2018. Dostupné také z: https://cs.wikipedia.org/w/index.php?title=Genotyp&oldid=16432335. [Online; navštíveno 10. 05. 2020].

14. WIKIPEDIE. *Sekvenování DNA — Wikipedie: Otevřená encyklopedie*. 2020. Dostupné také z: https://cs.wikipedia.org/w/index.php?title=Sekvenov%C3%A1n%C3%AD_DNA&oldid=18351282. [Online; navštíveno 4. 05. 2020].

15. PEREIRA, Rute; OLIVEIRA, Jorge; SOUSA, Mário. *Bioinformatics and Computational Tools for Next-Generation Sequencing Analysis in Clinical Genetics* [online]. Basel, Switzerland: MDPI, 2020 [cit. 2020-05-03]. Dostupné z: https://www.mdpi.com/2077-0383/9/1/132/pdf.

16. *Využití molekulárních markerův systematice a populační biologii rostlin* [online]. Praha: Katedra botaniky Přírodovědecké fakulty Univerzity Karlovy v Praze, 2013 [cit. 2020-05-04]. Dostupné z: https://botany.natur.cuni.cz/fer/markers/Markery11-NGS.pdf.

17. *NGS* [online]. USA: Bioneer [cit. 2020-05-04]. Dostupné z: https://eng.bioneer.com/images/products/ngs/ngs_WES.png.

18. NOVOTNÁ, Marcela. *Princip NGS metody* [online]. Hradec Králové: GENERI BIOTECH, 2018 [cit. 2020-05-04]. Dostupné z: https://www.generi-biotech.com/cs/princip-ngs-metody/.

19. *Zpracování dat z vysokokapacitního DNA sekvenování pro studium variability genomu a transkriptomu* [online]. Praha: Univerzita Karlova, 2018 [cit. 2020-05-04]. Dostupné z: https://is.cuni.cz/webapps/zzp/download/140067532/?lang=cs.

20. *Samtools* [online]. Samtools, 2019 [cit. 2020-05-04]. Dostupné z: http://www.htslib.org/doc/samtools.html.

21. [Online]. Samtools, 2019 [cit. 2020-05-04]. Dostupné z: http://www.htslib.org/doc/bcftools.html.

22. *Genome Analysis Toolkit* [online]. Cambridge (Massachuttes): Broad Institute [cit. 2020-05-04]. Dostupné z: https://gatk.broadinstitute.org/hc/en-us.

23. *FreeBayes* [online]. Praha: CESNET, 2017 [cit. 2020-05-04]. Dostupné z: https://wiki.metacentrum.cz/wiki/FreeBayes.

24. *Variant identification and analysis* [online]. UK: EMBL-EBI [cit. 2020-05-04]. Dostupné z: https://www.ebi.ac.uk/training/online/sites/ebi.ac.uk.training.online/files/GenVar_Fig_CRAM_file.png.

25. *The Variant Call Format Specification* [online]. Samtools organisation a repositories, 2020 [cit. 2020-05-04]. Dostupné z: http://samtools.github.io/hts-specs/VCFv4.3.pdf.

26. WIKIPEDIA CONTRIBUTORS. *Variant Call Format — Wikipedia, The Free Encyclopedia*. 2020. Dostupné také z: https://en.wikipedia.org/w/index.php?title=Variant_Call_Format&oldid=949428862. [Online; návštíveno 4. 05. 2020].

27. LUTZ, Jérôme. *DNA File Format Overview* [online]. SynBio.INFO [cit. 2020-05-04]. Dostupné z: http://synbio.info/display/synbio/DNA+File+Format+Overview.

28. *DNA Sequence formats* [online]. Germany: Genomatix [cit. 2020-05-04]. Dostupné z: https://www.genomatix.de/online_help/help/sequence_formats.html.

29. MARTÍNKOVÁ, Natália. *Sekvenování genomu* [online]. Brno: Institut biostatistiky a analýz Lékařské fakulty Masarykovy univerzity [cit. 2020-05-04]. Dostupné z: https://portal.matematickabiologie.cz/index.php?pg=analyza-genomickych-a-proteomickych-dat--analyza-sekvenci-dna--sekvenovani-genomu.

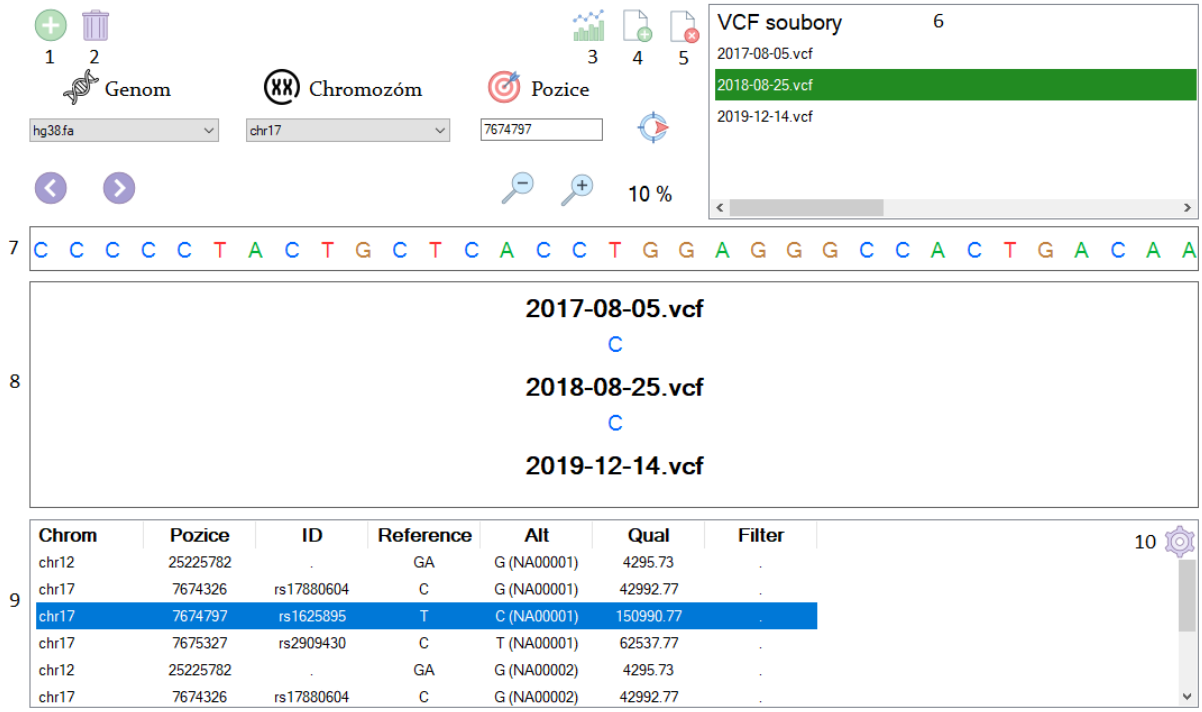
30. WIKIPEDIA CONTRIBUTORS. *FASTQ format — Wikipedia, The Free Encyclopedia*. 2020. Dostupné také z: https://en.wikipedia.org/w/index.php?title=FASTQ_format&oldid=953714170. [Online; návštíveno 4. 05. 2020].

31. WIKIPEDIE. *Fylogenetický strom — Wikipedie: Otevřená encyklopedie*. 2020. Dostupné také z: https://cs.wikipedia.org/w/index.php?title=Fylogenetick%C3%BD_strom&oldid=18398905. [Online; navštíveno 4. 05. 2020].

32. WIKIPEDIE. *C Sharp — Wikipedie: Otevřená encyklopedie*. 2020. Dostupné také z: https://cs.wikipedia.org/w/index.php?title=C_Sharp&oldid=18389885. [Online; navštíveno 6. 05. 2020].

33. *What is the Microsoft .NET Architecture?* [Online]. Progress [cit. 2020-05-06]. Dostupné z: https://www.progress.com/faqs/datadirect-ado-net-faqs/what-is-the-microsoft-net-architecture.

A Návod k obsluze aplikace



Obrázek 21: Hlavní okno aplikace

1. tlačítko pro přidání nového genomu
2. tlačítko pro odebrání aktuálně vybraného genomu
3. tlačítko pro zobrazení vývoje v časové řadě
4. tlačítko pro přidání VCF souboru
5. tlačítko pro odebrání aktuálně vybraného VCF souboru
6. seznam dostupných VCF souborů
7. panel pro zobrazení sekvencí genomu
8. panel pro zobrazení variant genových mutací
9. seznam variant vybraného VCF souboru (při dvojkliku na variantu v seznamu se daná varianta zobrazí)
10. tlačítko pro zobrazení filtru

A.1 První spuštění

1. Před prvním spuštěním je nutné si stáhnout referenční genom z následujícího odkazu ftp://hgdownload.soe.ucsc.edu/goldenPath/hg38/bigZips/latest/hg38.fa.gz.

2. Jakmile máte genom stáhnutý, tak můžete aplikaci spustit. Tento genom si přidejte tlačítkem pro přidání genomu. Nejprve vás aplikace vyzve k vybrání souboru ve formátu FASTA, to vyberte soubor s genomem, poté po vás bude chtít aplikace i indexový soubor. Ten najdete v přílohách pod názvem hg38.fa.fai. VCF soubory pro otestování taktéž najdete v příloze.

3. Pokud byste si chtěli zobrazit jiný genom, nejspíše u něj nebude indexový soubor. Tento soubor lze vygenerovat na operačním systému Linux následujícím příkazem: samtools faidx nazev_souboru.